

# VIZSGA FELADATSOR SZOFTVERTECHNOLÓGIA

c. tárgyból

2015. január 13.

**Az első lapon található feladatok megoldására 30 perc áll rendelkezésére. Az elérhető 24 pontból minimum 14 pontot kell kapnia ahhoz, hogy a második lapon szereplő feladatokra adott megoldásait értékeljük.**

*A tesztkérdésekre adott rossz válasz esetében pontot veszít, de feladatonként a total pontszám  $\geq 0$*

A feladatsor mentes az aktuálpolitikai témákra való utalásoktól és nem célja, hogy az államigazgatás bármely szervébe vetett közbizalmat megingassa, a tisztviselők iránti közmegebecsülést hátrányosan befolyásolja. A feladatsor szereplői nem valós személyek és bármilyen hasonlóság a valósággal csak a véletlen műve.

1. Jelölje (karikázza be) az állítások igazságtartalmát, ha feltesszük, hogy szabványos Java nyelvet használunk! (8 pont)

- I H** exceptionnek nem lehet *private* metódusa.
- I H** ha egy *x* objektum *package* láthatóságú metódusait meghívhatja egy másik osztályú *y* objektum, akkor *y* meghívhatja *x* *protected* metódusait is.
- I H** *RuntimeException* csak a virtuális gép beépített osztályai dobhatnak.
- I H** ha egy *FileReader*t egy *BufferedReader*be csomagolunk, akkor a *BufferedReader* bezárása után (*close* metódus meghívása) a *FileReader*t is be kell zárunk.
- I H** a *finally* blokkot a *garbage collector* hajtja végre.
- I H** *wait* metódust csak azon az objektumon lehet hívni, aminek a hívást végrehajtó szál a monitorában tartózkodik.
- I H** *finally* blokkból szabad kivételt dobni
- I H** ha van két szálunk, akkor a *join* metódusaikat csak az elindításuk sorrendjében szabad meghívni.

2. Az alábbi táblázatban adja meg a Rational Unified Process (RUP) fő munkafolyamatait és nevezze meg a hozzá tartozó nézetet. (5 pont)

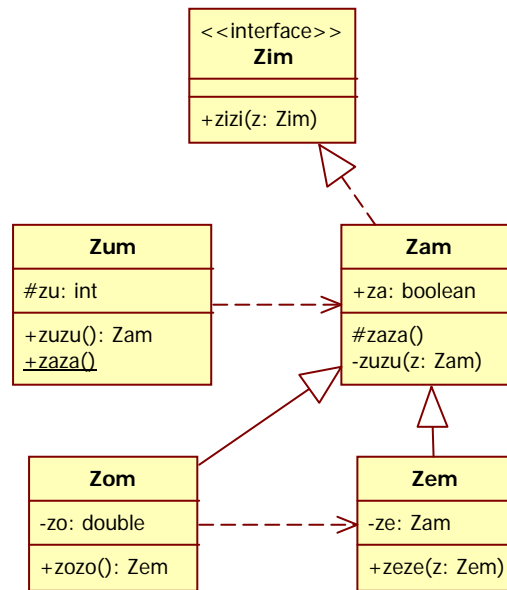
RUP fő munkafolyamatok/ RUP Process Workflows	Nézetek/Views
<b>Requirement</b>	<b>Use-case</b>
<b>Analysis</b>	<b>Logical</b>
<b>Design</b>	<b>Component</b>
<b>Implementation</b>	<b>Process</b>
<b>Deployment</b>	<b>Deployment</b>

3. Készítse el azt az állapottáblát, amely megfelel az alábbi DTD-vel definiált adatszerkezetnek ! Kötőjellel ( - ) jelölje, ha egy elem, egy állapotban nem következhet! (3 pont)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE x [
  <!ELEMENT x ((a+|b+),c)*>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT b (#PCDATA)>
  <!ELEMENT c (#PCDATA)>
]>
```

	a	b	c
1	2	3	-
2	2	-	1
3	-	3	1

4. Az alábbi UML2 diagram alapján - a kulcs felhasználásával - jellemezze az állításokat ! (8 pont)

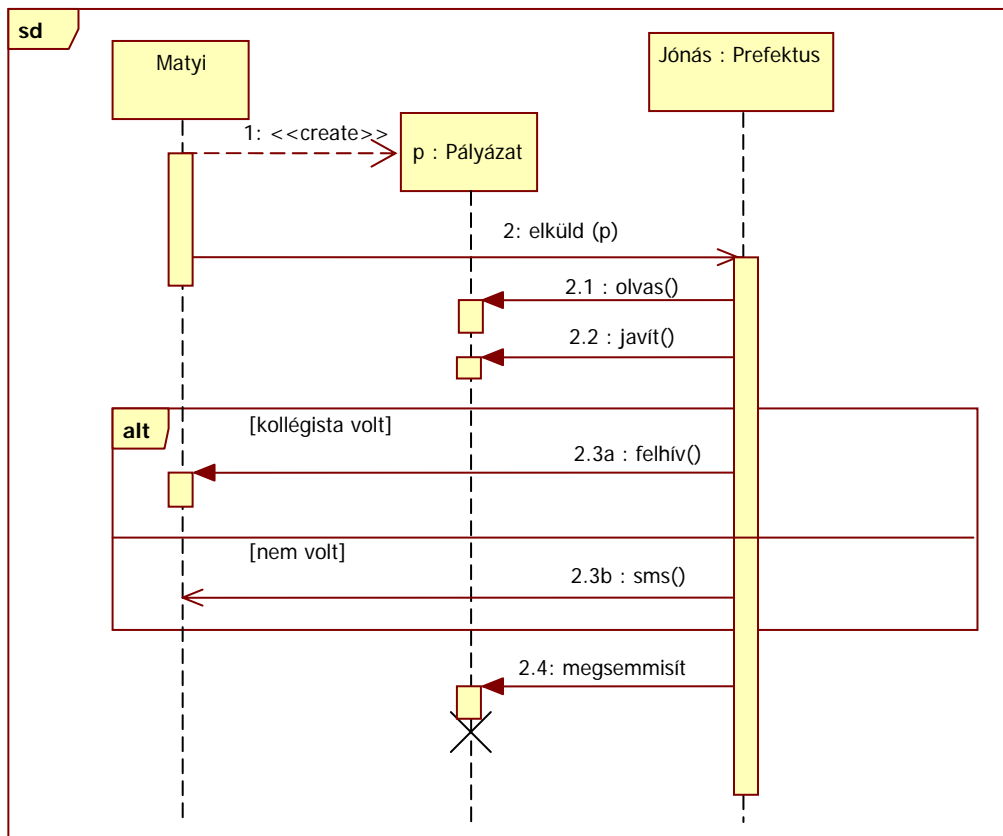


- A - csak az első tagmondat igaz (+ -)
- B - csak a második tagmondat igaz (- +)
- C - mindkét tagmondat igaz, de a következtetés hamis (+ + -)
- D - mindkét tagmondat igaz és a következtetés is helyes (+ + +)
- E - egyik tagmondat sem igaz (- -)

- [D] Van olyan **zaza** függvény, amely nem módosíthatja a **za** attribútumot, mert **za** nem statikus.
- [C] **Zum zuzu** függvénye nem példányosíthat **Zem** objektumot, mert **Zem** nem függ **Zum**-tól.
- [A] **Zom zozo** függvénye példányosíthat **Zom** objektumot, mert **Zom** a **Zem** leszármazottja.
- [A] **Zum zaza** függvénye nem módosíthatja a **zu** attribútum értékét, mert **zu** privát.
- [E] **Zom zizi** függvénye nem hívhatja meg **Zem zizi** függvényét, mert egyiknek sincs **zizi** függvénye.
- [C] **Zem zeze** függvénye nem kaphat paraméterül **Zom** objektumot, mert **Zom** függ **Zem**-től.
- [A] **Zam zuzu** függvénye meghívhatja a paraméterül kapott **Zom** típusú objektum **zaza** függvényét, mert **zaza** absztrakt.
- [D] **Zam zaza** függvénye nem hívhat meg minden **zuzu** függvényt, mert **Zam** nem ismeri **Zum**-ot.

5. Készítsen UML2 szekvenciadiagramot az alábbi történet alapján! (8 pont)

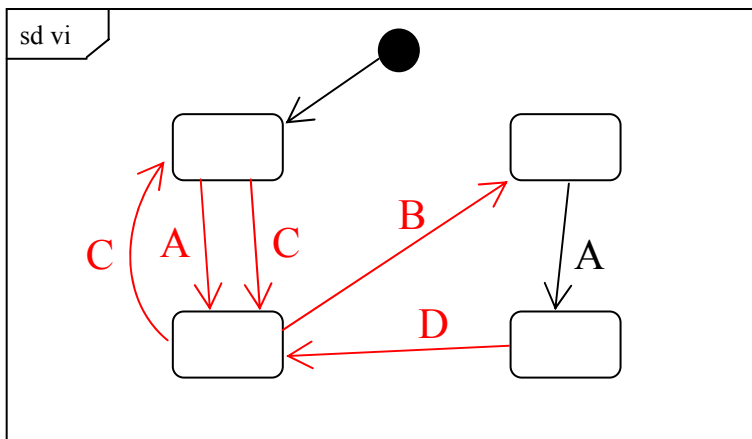
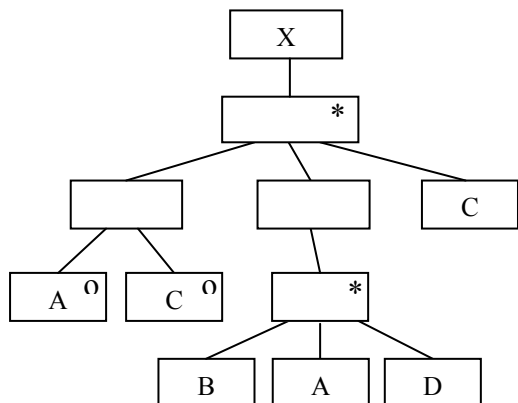
Nernia egy eldugott szegletében, Lopószenttökön Mutyi Matyi szeretne szotyolázót nyitni. Ezért készít egy pályázati anyagot, amit elküld Gázár Jónásnak, a prefektusnak. A prefektus elolvassa a pályázatot, majd kijavítja a helyesírási hibákat. Ha Matyi kollégista volt, akkor telefonon értesíti a sikerről, ha nem volt kollégista, akkor SMS-ben tájékoztatja. A prefektus az eljárás végén a pályázati anyagot (az adminisztráció egyszerűsítése céljából) megsemmisíti.



6. Adja meg, hogy az alábbi UML2 kollekció-jellemzők definiálása esetén melyik *java.util*-beli kollekció-interfészt használhatjuk! Adjon meg mindegyikhez egy tipikus *java.util*-beli megvalósítást is! (5 pont)

UML	Java util interfész	Java util megvalósítás
qualified	<b>Map</b>	<b>HashMap, Hashtable</b>
qualified, ordered	<b>SortedMap,</b>	<b>TreeMap</b>
unique	<b>Set</b>	<b>HashSet</b>
ordered	<b>List</b>	<b>ArrayList, Vector</b>
unique, ordered	<b>SortedSet</b>	<b>TreeSet</b>

7. Egészítse ki az UML2 állapotgépet úgy, hogy az ELH-nak megfelelő szerkezetet írja le! A kiegészítés során az **állapotgráfon új éleket vehet fel és az élekhez eseményeket írhat**. Bármilyen egyebet az ELH-n vagy az állapotgráfon változtatni tilos! CSAK a nyomtatott ábrába berajzoltakat értékeljük!(7 pont)  
 Javaslat: Készítsen vázlatot a harmadik lapon és CSAK a végső változatot rajzolja be a nyomtatott ábrába!

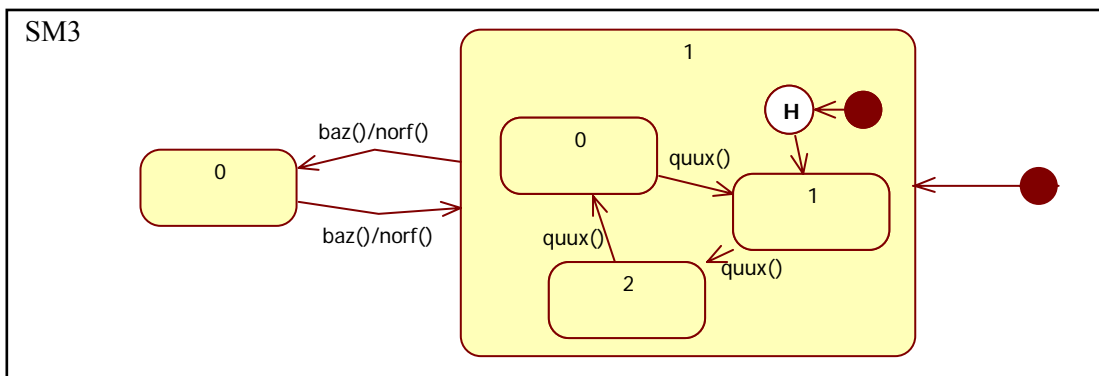


8. Rajzoljon UML2 állapotgépet az alábbi Java osztályhoz! (6 pont)

```
public class SM3 {
    private int x = 1;
    private int y = 1;
    private static final int p[] = { 1, 0 };

```

```
    private int s() { return (x+y)%3; }
    public void baz() { y = p[y]; norf(); }
    public void quux() { if (y == 1) x = s(); }
    public void norf() {}
}
```



Eredmények értékelése:

Pontszám	Osztályzat
21 -	2
28 -	3
35 -	4
42 -	5